CS 190
Testing Assignment
Handout #7
March 17, 1999

This assignment is intended to introduce you to the joys of testing and debugging. The goal is to evaluate how valuable two testing strategies are. Keep track of how long you spend on steps 3) and 5).

Each person in the class should:

1) Write some code. Choose one C++ file with at least 250 lines long to work with. Write the code and get it into a testable state. Generally, that means that it compiles and that you can get to the first line of code in each routine in the debugger.

2) Checkpoint the code. You can either make a copy or check it in to source control and "tag" it. By this time your source control person will have given you instructions on how to "tag" and "diff" files.

3) Step through each line of code in dbx. Follow the guidelines from Writing Solid Code chapter 4 on how to do this. The idea is to check variable values and other state information before and after execution of each line and verify that the line did what you expected. You will find bugs. Fix them. Continue the process. Whatever lines you change or add you will need to step through *again*.

4) When every line has been successfully stepped through, checkpoint the code again. Tag that version as well. This completes the first testing strategy.

5) Now do a code review with someone else in the class. Explain the structure of the code and what each line of code is doing. Ask your reviewer to be really certain that they understand what the code is doing. Augment your comments where they are confusing. Fix any bugs that you find. As in step 3), use the debugger to step through each line of code that you add or modify.

6) When every line has been successfully reviewed, checkpoint the code again and tag the file. This completes the second testing strategy.

7) You will now have three versions of your code. Hand in the following: a) the final version, b) both pairwise diffs between subsequent versions (1 vs. 2 and 2 vs. 3), c) the amount of time you spent on steps 3) and 5), and d) your opinion of the value of the two testing strategies. Remember, your opinion can't be wrong, only missing.